

**Amendments to the Claims**

1. (currently amended) A method of simulating application software, comprising:  
providing a user interface introducing a problem to be solved and tools to be used to solve the problem, wherein the problem is tied to at least one core curriculum subject matter;  
presenting a generic application user interface having elements common to several different software packages of a particular type of software application to a user across a network;  
receiving user inputs during an interaction with the user interface; and  
evaluating performance of the user for the type of software application based upon the user inputs.
2. (original) The method of claim 1, the method comprising providing feedback to the user on the performance of the user.
3. (original) The method of claim 2, providing feedback further comprising:  
informing the user of a wrong input; and  
presenting the user with an opportunity to try again.
4. (original) The method of claim 1, providing feedback further comprising  
informing the user of a wrong input;  
providing a hint to the user; and  
presenting the user with an opportunity to try again.
5. (original) The method of claim 1, providing feedback further comprising:  
informing the user of a wrong input; and  
demonstrating to the user the correct input.
6. (previously presented) The method of claim 1, the method comprising collecting all of the user inputs and evaluations of the user inputs and generating an evaluation report of the performance of the user for the particular type of software application.

7. (original) The method of claim 1, presenting a generic user interface for a particular application type further comprising presenting a generic user interface for an application selected from the group comprising: a spreadsheet, a word processor, and a presentation application.
8. (currently amended) A method of providing a user interface, comprising:  
providing an introduction to a problem for a user, wherein the problem is tied to at least one core curriculum subject matter;  
identifying tools to solve the problem, wherein the tools include at least one software application of a particular type;  
instructing the user on concepts and tools to be used in a solution;  
interacting with the user using the tools;  
displaying the solution; and  
providing the user with a summary of the problem and solution.
9. (original) The method of claim 8, interacting with the user further comprising receiving a user input and storing the user input for further evaluation.
10. (original) The method of claim 8, interacting with the user further comprising receiving and evaluating a user input.
11. (original) The method of claim 8, interacting with the user further comprising providing feedback.
12. (original) The method of claim 11, providing feedback further comprising indicating that the user made a correct input.
13. (previously presented) The method of claim 11, providing feedback further comprising indicating that the user made an incorrect input and displaying a region in a window in which the user may make another input.

14. (currently amended) A method of providing an integrated technology learning system, comprising:

establishing technology objectives for an instructional unit;

identifying core curriculum subject matter components related to the technology objectives;

providing a theme and characters for the unit based upon the core curriculum subject matter components;

determining a framework for problem solving;

determining common elements of a user interface for at least one particular type of software application; and

adding simulation functionality for that particular type.

15. (original) The method of claim 14, adding simulation functionality further comprising:

recording user inputs in response to prompts;

recording a environment from which the user input is recorded; and

storing the user inputs and the environment.

16. (currently amended) An instructional management system, comprising:

at least one instructional unit having at least one task for which a student is required to provide an input, wherein the instructional unit is tied to at least one core curriculum subject matter;

user interface having elements common to several different software packages of a particular type of software application to simulate the software application having a region to allow the student to provide the input;

a memory in which to record the student input;

a report generator to allow an instructor to access results of the inputs and to provide an evaluation of student performance with regard to the software application.

17. (currently amended) An article of machine-readable media containing instructions that, when executed, cause the machine to:

provide a user interface introducing a problem to be solved and tools to be used to solve the problem, wherein the problem is tied to at least one core curriculum subject matter;

present a generic application user interface having elements common to several different software packages of a particular type of application to a user across a network;

receive user inputs during an interaction with the user interface; and

evaluate performance of the user with regard to the type of application based upon the user inputs.

18. (original) The article of claim 17, the article containing instructions that, when executed, cause the machine to provide feedback to the user on the performance of the user.

19. (currently amended) An article of machine-readable media containing instructions that, when executed, cause the machine to:

provide an introduction to a problem for a user, wherein the problem is tied to at least one core curriculum subject matter;;

identify tools to solve the problem, wherein the tools include at least one software application of a particular type;

instruct the user on concepts and tools to be used in a solution;

interact with the user using the tools;

display the solution; and

provide the user with a summary of the problem and solution.